CS224N NLP

Christopher Manning
Spring 2010

Borrows slides from Bob Carpenter, Dan Klein, Roger Levy, Josh Goodman, Dan Jurafsky

Five types of smoothing

- We'll cover
  - Add-δ smoothing (Laplace)
  - Simple interpolation
  - Good-Turing smoothing
  - Katz smoothing
  - Kneser-Ney smoothing
- Or less if we run out of time … and then you'll just have to read the textbook!
  - J&M: ch. 4.

Last Time's Quiz Question!

- Suppose I'm making a language model with a vocabulary size of 20,000 words
- In my training data, I saw the bigram *comes across* 10 times
  - 5 times it was followed by *as*
  - 5 times it was followed by other words (*like, less, again, most, in*)
- What is the MLE of P(*as*|*comes across*)?
- What is the add-1 estimate of P(*as*|*comes across*)?

How Much Mass to Withhold?

- Remember the key discounting problem:
  - What count should r^* should we use for an event that occurred r times in N samples?
  - r is too big
- Idea: held-out data [Jelinek and Mercer]
  - Get another N samples
  - See what the average count of items occuring r times is (e.g., doubletons on average might occur 1.78 times)
  - Use those averages as r^*
- Works better than fixing counts to add in advance

Backoff and Interpolation

- *Discounting* says, "I saw event X *n* times, but I will really treat it as if I saw it fewer than *n* times
- *Backoff* (and *interpolation*) says, "In certain cases, I will condition on less of my context than in other cases"
  - The sensible thing is to condition on less in contexts that you haven't learned much about
- **Backoff:** use trigram if you have it, otherwise bigram, otherwise unigram
- **Interpolation:** mix all three

Linear Interpolation

- One way to ease the sparsity problem for n-grams is to use less-sparse n-1-gram estimates
- General linear interpolation:

$$P(w \mid w_{-1}) = [1 - \lambda(w, w_{-1})]\hat{P}(w \mid w_{-1}) + [\lambda(w, w_{-1})]P(w)$$

  - Having a single global mixing constant is generally not ideal:

$$P(w \mid w_{-1}) = [1 - \lambda]\hat{P}(w \mid w_{-1}) + [\lambda]P(w)$$

    - But it actually works surprisingly well – simplest competent approach
  - A better yet still simple alternative is to vary the mixing constant as a function of the conditioning context

$$P(w \mid w_{-1}) = [1 - \lambda(w_{-1})]\hat{P}(w \mid w_{-1}) + \lambda(w_{-1})P(w)$$

Held-Out Data

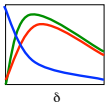
- Important tool for getting models to generalize:

  - When we have a small number of parameters that control the degree of smoothing, we set them to maximize the (log-)likelihood of held-out data

$$LL(w_1 \ldots w_n \mid M(\lambda_1 \ldots \lambda_k)) = \sum_i \log P_{M(\lambda_1 \ldots \lambda_k)}(w_i \mid w_{i-1})$$

  - Can use any optimization technique (line search or EM usually easiest)
- Example:

$$P(w \mid w_{-1}) = [1-\lambda]\hat{P}(w \mid w_{-1}) + [\lambda]P(w)$$



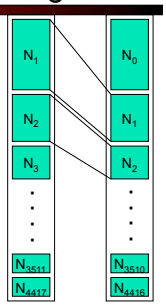
Good-Turing smoothing intuition

- Imagine you are fishing
- You have caught
  - 10 carp, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel = 18 fish
- How likely is it that next species is new (i.e. catfish or bass)
  - 3/18
- Assuming so, how likely is it that next species is trout?
  - Must be less than 1/18

[Slide adapted from Josh Goodman]

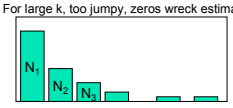
Good-Turing Reweighting I

- We'd like to not need held-out data (why?)
- Idea: leave-one-out validation
  - Take each of the c training words out in turn
  - c training sets of size c-1, held-out of size 1
  - What fraction of held-out words are unseen in training?
    - N_1/c
  - What fraction of held-out words are seen k times in training?
    - $(k+1)N_{k+1}/c$
  - So in the future we expect $(k+1)N_{k+1}/c$ of the words to be those with training count k
  - There are N_k words with training count k
  - Each should occur with probability:
    - $(k+1)N_{k+1}/c/N_k$
  - …or expected count $(k+1)N_{k+1}/N_k$

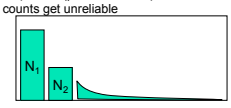


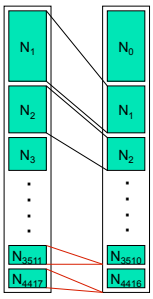
Good-Turing Reweighting II

- Problem: what about "the"? (say c=4417)
  - For small k, $N_k > N_{k+1}$
  - For large k, too jumpy, zeros wreck estimates



  - Simple Good-Turing [Gale and Sampson]: replace empirical N_k with a best-fit power law once count counts get unreliable





Good Turing calculations

	unseen (bass or catfish)	trout
c	0	1
MLE p	$p = \frac{0}{18} = 0$	$\frac{1}{18}$
c^*		$c^*(\text{trout}) = 2 \times \frac{N_2}{N_1} = 2 \times \frac{1}{3} = .67$
GT p^*_{GT}	$p^*_{GT}(\text{unseen}) = \frac{N_1}{N} = \frac{3}{18} = .17$	$p^*_{GT}(\text{trout}) = \frac{.67}{18} = \frac{1}{27} = .037$

Good-Turing Reweighting III

- Hypothesis: counts of k should be $k^* = (k+1)N_{k+1}/N_k$

Count in 22M Words	Actual c* (Next 22M)	GT's c*
1	0.448	0.446
2	1.25	1.26
3	2.24	2.24
4	3.23	3.24
Mass on New	9.2%	9.2%

- Katz Smoothing
  - Extends G-T smoothing into a backoff model from higher to lower order contexts
  - Use G-T discounted *bigram* counts (roughly – Katz left large counts alone)
  - Whatever mass is left goes to empirical unigram

$$P_{KATZ}(w \mid w_{-1}) = \frac{c^*(w, w_{-1})}{\sum_w c(w, w_{-1})} + \alpha(w_{-1})\hat{P}(w)$$

Intuition of Katz backoff + discounting

- How much probability to assign to all the zero trigrams?
  - Use GT or some other discounting algorithm to tell us
- How do we divide that probability mass among different words in the vocabulary?
  Use the (*n* – 1)-gram estimates to tell us
- What do we do for the unigram words not seen in training (i.e., not in our vocabulary)?
  - The problem of Out Of Vocabulary = OOV words
    - Important, but messy … mentioned at the end

Kneser-Ney Smoothing I

- Something's been very broken all this time
  - Shannon game: There was an unexpected ____?
    - delay?
    - Francisco?
  - "Francisco" is more common than "delay"
  - … but "Francisco" always follows "San"
- Solution: Kneser-Ney smoothing
  - In the back-off model, we don't want the unigram probability of w
  - Instead, probability given that we are observing a novel continuation
  - Every bigram type was a novel continuation the first time it was seen

$$P_{CONTINUATION}(w) = \frac{|\{w_{-1} : c(w, w_{-1}) > 0\}|}{|(w, w_{-1}) : c(w, w_{-1}) > 0|}$$

Kneser-Ney Smoothing II

- One more aspect to Kneser-Ney:
  - Look at the GT counts:

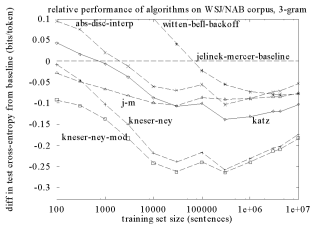
Count in 22M Words	Actual c* (Next 22M)	GT's c*
1	0.448	0.446
2	1.25	1.26
3	2.24	2.24
4	3.23	3.24

- Absolute Discounting
  - Save ourselves some time and just subtract 0.75 (or some d)
  - Maybe have a separate value of d for very low counts

$$P_{KN}(w \mid w_{-1}) = \frac{c(w, w_{-1}) - D}{\sum_{w'} c(w', w_{-1})} + \alpha(w_{-1}) P_{CONTINUATION}(w)$$

What Actually Works?

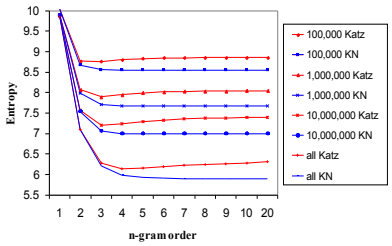
- Trigrams:
  - Unigrams, bigrams too little context
  - Trigrams much better (when there's enough data)
  - 4-, 5-grams usually not worth the cost (which is more than it seems, in dynamic programming searches)
- Good-Turing-like methods for count adjustment
  - Absolute discounting, Good-Turing, held-out estimation, Witten-Bell
- Kneser-Ney equalization for lower-order models
- See [Chen+Goodman] reading for tons of graphs!



[Graph from Joshua Goodman]

Data >> Method?

- Having more data is always good…



- … but so is picking a better smoothing mechanism!
- *n* > 3 often not worth the cost … but is with enough data

Google N-Gram Release

All Our N-gram are Belong to You

By Peter Norvig - 8/03/2006 11:26:00 AM

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects, such as statistical machine translation, speech recognition, spelling correction, entity detection, information extraction, and others. While such models have usually been estimated from training

to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Google N-Gram Release

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensible 40
- serve as the individual 234

Beyond N-Gram LMs

- Caching Models
  - Recent words more likely to appear again

$$P_{CACHE}(w \mid history) = \lambda P(w \mid w_{-1}w_{-2}) + (1-\lambda)\frac{c(w \in history)}{|history|}$$

  - Can be disastrous in practice for speech (why?)
- Skipping Models

$$P_{SKIP}(w \mid w_{-1}w_{-2}) = \lambda_1 \hat{P}(w \mid w_{-1}w_{-2}) + \lambda_2 P(w \mid w_{-1}\_\_) + \lambda_3 P(w \mid \_\_ w_{-2})$$

- Clustering Models: condition on word classes when words are too sparse
- Trigger Models: condition on bag of history words (e.g., maxent)
- Structured Models: use parse structure (we'll see these later)
- Language Modeling toolkits
  - SRILM
  - CMU-Cambridge LM Toolkit
  - IRST LM Toolkit

Unknown words: Open versus closed vocabulary tasks

- If we know all the words in advance
  - Vocabulary V is fixed
  - **Closed vocabulary task.** Easy
    - Common in speech recognition.
- Often we don't know the set of all words
  - Out Of Vocabulary = OOV words
  - **Open vocabulary task**
- Instead: create an unknown word token <UNK>
  - Training of <UNK> probabilities
    - Create a fixed lexicon L of size V. [Can we work out right size for it??]
    - At text normalization phase, any training word not in L changed to <UNK>
    - There may be no such instance if L covers the training data
  - Now we train its probabilities
    - If low counts are mapped to <UNK>, we may train it like a normal word
    - Otherwise, techniques like Good-Turing estimation are applicable
  - At decoding time
    - If text input: Use UNK probabilities for any word not in training

Practical Considerations

- The unknown word symbol <UNK>
  - In many cases, open vocabularies use multiple types of OOVs (e.g., numbers & proper names)
  - For the programming assignment:
    - OK to assume there is only one unknown word type, UNK
    - UNK may be quite common in new text!
    - UNK stands for all unknown word types (define probability event model thus – it's a union of basic outcomes)
  - To model the probability of individual new words occurring, you can use spelling models for them, but people usually don't
- Numerical computations
  - We usually do everything in log space (log probabilities)
    - Avoids underflow
    - (also adding is faster than multiplying)

$$p_1 \times p_2 \times p_3 \times p_4 = \exp(\log p_1 + \log p_2 + \log p_3 + \log p_4)$$

Today's Quiz Question!

- Suppose I'm making a language model with a vocabulary size of 20,000 words
- In my training data, I saw the bigram *comes across* 10 times [these are authentic counts from a small corpus, BTW]
  - 5 times it was followed by *as*
  - 5 times it was followed by other words (*like, less, again, most, in*)
- The next time I see *comes across:*
  - According to Good-Turing smoothing, what is the probability of seeing a different, previously unseen word following it?
  - Using absolute discounting with D = 0.75, what is the probability of seeing *as* after it?

Machine Translation: Word alignment models

Christopher Manning
CS224N
[Based on slides by Kevin Knight, Dan Klein, Dan Jurafsky]

"When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.' " – Warren Weaver, March 1947

"… as to the problem of mechanical translation, I frankly am afraid that the [semantic] boundaries of words in different languages are too vague … to make any quasi-mechanical translation scheme very hopeful." – Norbert Wiener, April 1947

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: farok crrrok hihok yorok clok kantok ok-yurp

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: farok crrrok hihok yorok clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: farok crrrok hihok yorok clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok **farok** ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok **farok** izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: farok crrrok hihok yorok clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok **farok** ororok lalok sprok izok enemok . 7b. wat **jjat** bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok **farok** izok stok . 5b. totat **jjat** quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok hihok yorok clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok **farok** ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok **farok** izok stok . 5b. totat jjat quat cat .	11a. lalok nok **crrrok** hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

???

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok hihok **yorok** clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok **hihok** ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok **hihok** yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok **hihok** mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok **hihok** yorok **clok** kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok **hihok** ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok **yorok** ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok **hihok** **yorok** zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok **hihok** mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok **hihok** **yorok** clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok **clok** . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok **hihok** **yorok** clok kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok **hihok yorok clok** kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

process of elimination

Centauri/Arcturan [Knight, 1997]

Your assignment, translate this to Arcturan: **farok** crrrok **hihok yorok clok** kantok ok-yurp

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

cognate?

Centauri/Arcturan [Knight, 1997]

Your assignment, put these words in order: **{ jjat, arrat, mat, bat, oloat, at-yurp }**

1a. ok-voon ororok sprok . 1b. at-voon bichat dat .	7a. lalok farok ororok lalok sprok izok enemok . 7b. wat jjat bichat wat dat vat eneat .
2a. ok-drubel ok-voon anok plok sprok . 2b. at-drubel at-voon pippat rrat dat .	8a. lalok brok anok plok nok . 8b. iat lat pippat rrat nnat .
3a. erok sprok izok hihok ghirok . 3b. totat dat arrat vat hilat .	9a. wiwok nok izok kantok ok-yurp . 9b. totat nnat quat oloat at-yurp .
4a. ok-voon anok drok brok jok . 4b. at-voon krat pippat sat lat .	10a. lalok mok nok yorok ghirok clok . 10b. wat nnat gat mat bat hilat .
5a. wiwok farok izok stok . 5b. totat jjat quat cat .	11a. lalok nok crrrok hihok yorok zanzanok . 11b. wat nnat arrat mat zanzanat .
6a. lalok sprok izok jok stok . 6b. wat dat krat quat cat .	12a. lalok rarok nok izok hihok mok . 12b. wat nnat forat arrat vat gat .

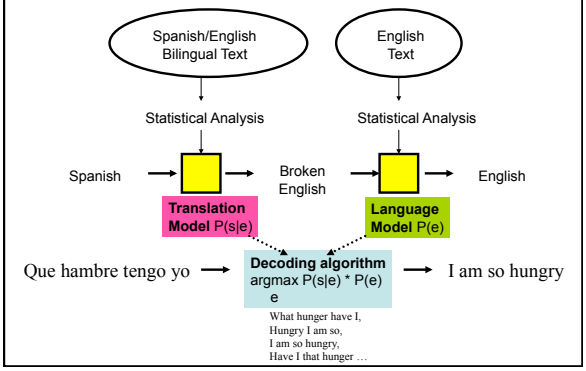
zero fertility

It's Really Spanish/English

Clients do not sell pharmaceuticals in Europe **=> Clientes no venden medicinas en Europa**

1a. Garcia and associates . 1b. Garcia y asociados .	7a. the clients and the associates are enemies . 7b. los clients y los asociados son enemigos .
2a. Carlos Garcia has three associates . 2b. Carlos Garcia tiene tres asociados .	8a. the company has three groups . 8b. la empresa tiene tres grupos .
3a. his associates are not strong . 3b. sus asociados no son fuertes .	9a. its groups are in Europe . 9b. sus grupos estan en Europa .
4a. Garcia has a company also . 4b. Garcia tambien tiene una empresa .	10a. the modern groups sell strong pharmaceuticals . 10b. los grupos modernos venden medicinas fuertes .
5a. its clients are angry . 5b. sus clientes estan enfadados .	11a. the groups do not sell zenzanine . 11b. los grupos no venden zanzanina .
6a. the associates are also angry . 6b. los asociados tambien estan enfadados .	12a. the small groups are not modern . 12b. los grupos pequenos no son modernos .

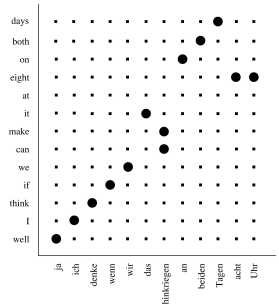
Statistical MT Systems



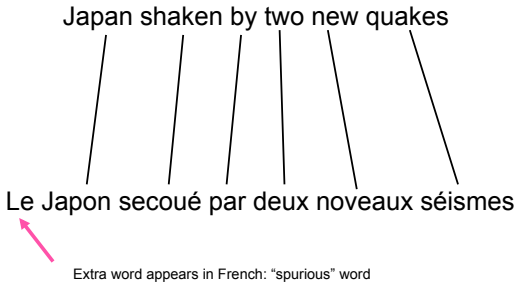
A division of labor

- Use of Bayes Rule ("the noisy channel model") allows a division of labor:
  - Job of the translation model P(E|S) is just to model how various Spanish words typically get translated into English (perhaps in a certain context)
    - P(E|S) doesn't have to worry about language-particular facts about English word order: that's the job of P(E)
  - The job of the language model is to choose felicitous bags of words and to correctly order them for English
    - P(E) can do bag generation: putting a bag of words in order:
      - E.g., hungry I am so → I am so hungry
- Both can be incomplete/sloppy

Word Alignment Examples: Grid

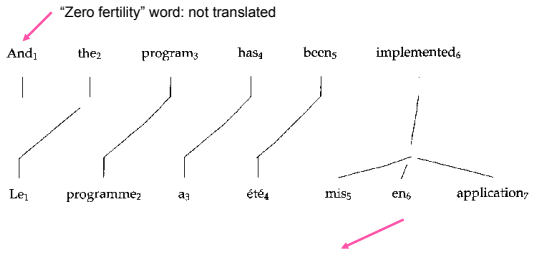


Word alignment examples: easy

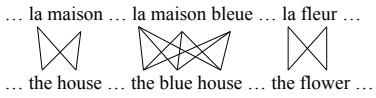


Extra word appears in French: "spurious" word

Alignments: harder



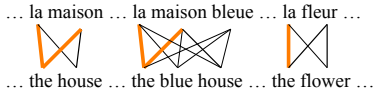
Statistical Machine Translation



All word alignments equally likely

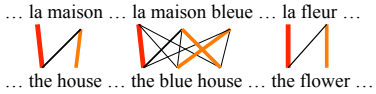
All P(french-word | english-word) equally likely

Statistical Machine Translation



"la" and "the" observed to co-occur frequently, so P(la | the) is increased.

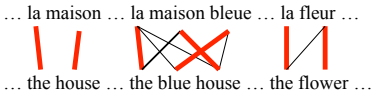
Statistical Machine Translation



"maison" co-occurs with both "the" and "house", but P(maison | house) can be raised without limit, to 1.0, while P(maison | the) is limited because of "la"

(pigeonhole principle)

Statistical Machine Translation

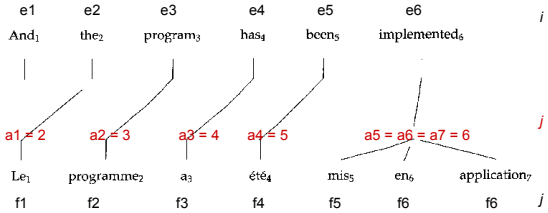


settling down after another iteration

That was the idea of IBM Model 1. For details, see the next slides and:
- "A Statistical MT Tutorial Workbook" (Knight, 1999).
- "The Mathematics of Statistical Machine Translation" (Brown et al, 1993)
- Software: GIZA++

Model 1 parameters

- $P(F|E) = \Pi_{(f,e)} P(f|e)$
- $P(f|e) = P(J|I) \Sigma_a P(f, a|e)$
- $P(f, a|e) = \Pi_j P(a_j = i) P(f_j|e_i) = \Pi_j [1/(I+1)] P(f_j|e_i)$



Model 1: Word alignment learning with Expectation-Maximization (EM)

- Start with $P(f^j|e^i)$ uniform, including $P(f^j|\text{NULL})$
- For each sentence
  - For each French position *j*
    - Calculate posterior over English positions $P(a_j|i)$

$$P(a_j = i \mid f, e) = \frac{P(f_j \mid e_i)}{\sum_{i'} P(f_j \mid e_{i'})}$$

    - Increment count of word f_j with word e_{a_j}
      - $C(f_j|e_i)$ += $P(a_j = i \mid f, e)$
- Renormalize counts to give probs $P(f^p \mid e^q) = \frac{C(f^p \mid e^q)}{\sum_{f^s} C(f^s \mid e^q)}$
- Iterate until convergence